This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

THIS PAGE BLANK (USPTO)

Journal of Electronic Imaging 7(3), 571–577 (July 1998).

XP-000771764

Block-matching algorithm based on dynamic adjustment of search window for low bit-rate video coding

Hwang-Seok Oh
Heung-Kyu Lee
Korea Advanced Institute of Science and Technology (KAIST)
Department of Computer Science
373-1, Kusong-Dong, Yusong-Gu
Taejon 305-701, Republic of Korea
E-mail: hsoh@casaturn.kaist.ac.kr

YN. 8403
P.D. 00/07/98
p. 551-577 = 7

Abstract. *A dynamic search window adjustment for block-matching algorithm (BMA) based on the block similarity is presented to reduce the computational complexity of full search BMA. The adjustment of the size of the search window is performed in three steps: (1) set a new search origin based on the block similarity and the displaced block difference (DBD), (2) adjust the size of search window in inverse proportion to block similarity, and (3) update the thresholds for accommodation to a given image sequence. The technique can be easily applied to full search BMA and several fast search algorithms to get more efficiency and to reduce a possibility of falling into a local minimum. Experimental results show that the proposed technique has a good MSE performance and reduces the number of search points substantially.* © *1998 SPIE and IS&T.*
[S1017-9909(98)02303-4]

1 Introduction

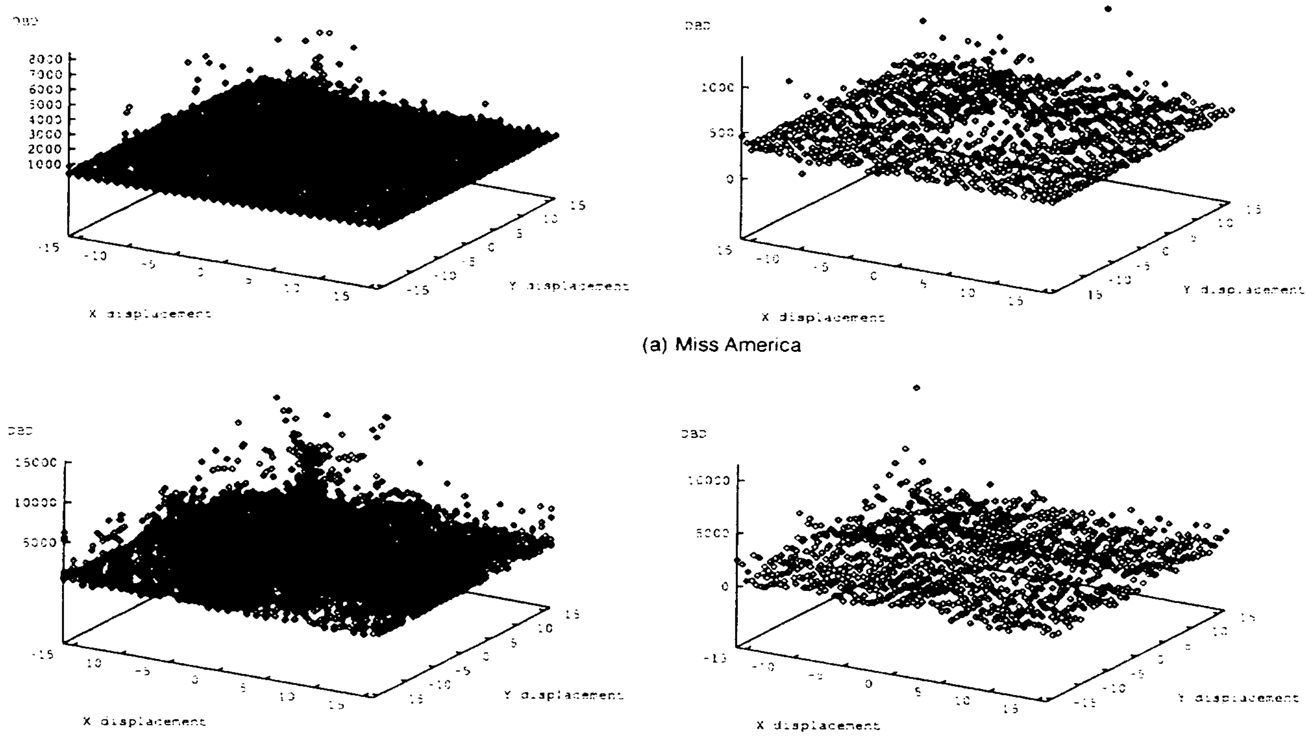
Motion compensated prediction plays a very important role in the efficient coding of video sequences. MPEG-1,2,[1] ITU-T H.261,[2] and H.263[3] adopted the method to reduce temporal redundancies which reside in successive frames. Most of the algorithms developed for motion estimation so far use a block-based technique called block-matching algorithm (BMA), that estimate the motion vector (MV) block-by-block. In BMA, a frame is divided into nonoverlapping blocks with $(N\times N)$ pixel size. A block of pixels (called a *current block*) in the current frame is compared with its corresponding blocks (called *candidate blocks*) within a search area of size $(N+2w)\times(N+2w)$ in the reference frame, where w is the maximum displacement of the MV. The MV of the current block is obtained when the best matched candidate block is found. The general approach for BMA is to use full search block-matching algorithm (FSBMA). In FSBMA, all possible $(2w-1)^2$ candidate blocks are compared to obtain the best matched block.[4]

Because of the intensive computation to get MV or displacement of the block that has the smallest distortion function of the matching criterion in FSBMA, many fast search algorithms such as three step search (TSS),[5] new three step search (NTSS),[6] 2D logarithm search,[7] one at a time search algorithm (OTS),[8] 1D full search,[9] cross search algorithm (CSA),[10] and parallel hierarchical 1D search (PHODS),[11] etc., have been investigated. But there is a critical problem with these techniques: falling into a local minimum, owing to the assumption that the distortion increases monotonically as the searched point moves away from the position of minimum distortion.

Another approach to reduce the computational complexity is an adjustment of the size of the search window w, which has been suggested in Refs. 12 and 13. In Ref. 12, a method to reduce the size of the search window in TSS according to the magnitude of the DBD was presented. The approach proposed in Ref. 13, the size of the search window is determined in proportion to the DBD of blocks. The method, called an adaptive adjustment of search window (AASW) for BMA, exploits the motion correlation of spatially neighboring blocks to determine the search origin and adjusts the search range according to the different motion content of the block. The scheme is performed in three stages: (1) set a search origin, (2) determine the size of search window, and (3) update the thresholds for classification of motion contents of block frame by frame. The search origin is determined with motion vectors of the adjacent blocks in the left, upper-left, and upper directions, together with zero displacement to predict a motion vector of the current block; then, a vector that has the minimum DBD is selected as a predicted motion vector. The location pointed to by the predicted motion vector is used as a search origin for FSBMA. After setting of the search origin, the size of the search area is determined by considering the DBD in a position of the new search origin. The DBD is used as an indicator of the degree of motion for a given block. It was proposed that the DBD be used as the starting search point as a criterion to identify the motion class of the

Paper IVC-08 received June 5, 1997; revised manuscript received Oct. 29, 1997; accepted for publication Apr. 24, 1998.
1017-9909/98/$10.00 © 1998 SPIE and IS&T.

BNSDOCID: <XP____771764A_I_>



(a) Miss America

(b) Carphone

Fig. 1 The distribution of the DBD and the mean DBD along the x,y components of MVs.

block. Three motion classes—low motion, medium motion, and high motion—are defined according to the magnitude of the DBD and their maximum displacements are set to $w/4$, $w/2$, w, respectively.

The search window adjustment proposed by Feng *et al.*[13] has some problems. First, they assume that the magnitude of MV is proportional to the magnitude of the DBD. That is, if the best matched candidate block in the reference block is far away from the search origin, then, the DBD becomes very large. But, with the real test images, we can see that there is no significant correlation between them (see Fig. 1). In Fig. 1, the DBD distribution and the mean DBD value obtained by FSBMA with two video sequences are plotted along the displacement of horizontal and vertical directions, respectively. At each location of the MV, the mean DBD values are similar without regard to the magnitude of the MV component. For example, the mean DBDs at location $(-15,-15)$, $(-15,15)$, $(15,-15)$, and $(15,15)$ are similar to the mean DBD at location $(0,0)$. In general, since the DBD is very large in the area of the motion boundary, the DBD is dependent on the content of the block rather than the magnitude of the MV. That is, the DBD is not significantly affected by the displacement of the MV. On the other hand, the adjacent blocks belong to the same moving region have very similar motion field. So the spatial correlation of adjacent blocks can be used for determination of the size of the search area to reduce the computational complexity.

In addition to the video sequences for the low bit-rate applications such as video phone or video conferencing are gentle, smooth, and vary slowly. Thus, the MV distribution of the best-matched block is center biased. Figure 1 shows two facts: (1) the MV distribution is center biased and (2) the DBD does not highly correlate to the magnitude of the MV. If the spatial correlation between adjacent blocks is used well, the mismatch of the DBD and magnitude of the MV is remedied and more reduction of computational complexity can be achieved by prediction of the initial MV by using motion displacement of an adjacent block.

In this paper, we propose a new BMA with dynamic adjustment of search window (DASW) to overcome the complexity of FSBMA. In the DASW algorithm, the new search origin for current block and the size of search area for MV are determined by considering block similarities, MV correlations of adjacent blocks, and their DBDs. Then, the general mechanism of FSBMA or several other fast search algorithms can be applied within the determined displacement.

This paper is organized as follows. In Sec. 2, we describe the detailed algorithm of the DASW to determine the size of the search window for each block. In Sec. 3, experimental results to compare with the performance of methods for search window adjustment are presented using several test video sequences for low bit-rate video applications such as video phone or video conferencing.

2 Dynamic Adjustment of the Search Window for BMA

In video sequences, especially for the low bit-rate applications such as video phone or video conferencing, the mo-

BNSDOCID: <XP____771764A__I_>

Algorithm : Dynamic Adjustment of Search Window(DASW)

step 1 : Segment the current frame using a simple split algorithm,

step 2 : For a given current block, compute block similarities between the block and upper, left, upper-left, upper-right blocks. Then, mark the block that has the largest similarity with the current block. The block similarity is calculated by using the segmentation information of the current frame as the ratio of the number of pixels in two blocks belong to the same region to the number of pixels in a block;

step 3 : Calculate the DBD of the current block at location pointed to by the MV of an adjacent block which has the largest block similarity(called DBD_{adj}) and at zero displaced location(called $DBD_{(0,0)}$);

Step 4 : Set a new search origin as either the position pointed to by MV of the adjacent block or zero displaced position, that has smaller DBD of the two DBD_{adj} and $DBD_{(0,0)}$;

Step 5 : Determine the search window size as follows:

Step 5.1 : If the new search origin is set in a location pointed to by MV of the adjacent block ($DBD_{(0,0)} > DBD_{adj}$), the size of search window is adjusted by considering block similarity, i.e. inversely proportional to block similarity;

step 5.2 : If the new search origin is set as zero displaced position($DBD_{(0,0)} \leq DBD_{adj}$), the size of search window is determined to be (block similarity) $* Max\{x, y\}$ MV component of the adjacent block } + $w *$ (1 − (block similarity)). Where, w is an initial displacement;

Step 6 : Perform BMA at new search origin with the adjusted displacement;

Step 7 : Update the threshold T_{seg} for segmentation by considering the number of matching blocks frame by frame;

Fig. 2 The DASW algorithm for determination of the size of a search window based on the block similarity and DBDs.

tion field is smooth and changes slowly frame by frame. The correlation between MVs for adjacent blocks is very high if each block belongs to the same object because an object spans several blocks. Also, as shown in Fig. 1, there are many blocks such that their MVs are near the search origin in video sequences.

Based on these facts, we present a new BMA with DASW, which exploits motion structures of objects to reduce the number of matching blocks (candidate blocks). The DASW algorithm is described briefly in Fig. 2. Because each motion displacement of the block is greatly related to the moving objects in successive video frames, if some blocks belong to the same object region, they have similar motion displacements and DBDs. In the DASW, we take advantage of motion structures of objects to determine the size of the search window by considering the block similarity which is computed using segmentation information of a given frame.

2.1 *Set New Search Origin*

In our approach, we consider the block similarity of spatially adjacent blocks to determine the size of the search window. In video sequences, if some adjacent blocks are contained in the same object, the motion structures of the blocks are very similar. Therefore, we can predict an initial MV as the MV of an adjacent block that is significantly related to the current block and refine the MV at that position with a smaller displacement. To determine the block similarity of adjacent blocks, we use the segmentation information of a given current frame. In general, segmented regions are expected to have homogeneous characteristics such as intensity and texture that are different in each region. These characteristics form the feature vectors that are used to discriminate one region from the other. The features are employed during the segmentation procedure in the checking region homogeneity. Many techniques for image segmentation are summarized in the literature.[17] In the DASW, we apply a simple method using a region splitting algorithm for segmentation.[17] The region splitting algorithm is a top-down approach and it starts with the assumption that the entire image is homogeneous. If this is not true, the image is split into four subimages. This splitting procedure is repeated recursively until homogeneous image regions are encountered. The homogeneity is checked whether the pixel difference in a region is greater than a given threshold T_{seq} or not.

In the DASW scheme, we use the segmented frame information together with MVs of adjacent blocks and their DBDs for setting the new search origin. First, a new search origin is selected among blocks which are displaced as much as zero or MV of its neighboring block by considering the block similarity. The block similarity is calculated using the segmentation information of the whole current frame as how many pixels in two blocks belong to the same region over the number of pixels in a block. A new search origin is set as either the zero displaced or the displacement as much as the MV of an adjacent block with the maximum block similarity. Among the two candidates, the displacement of the block that has smaller DBD is taken as the new search origin. Let $DBD_{(0,0)}$ be the DBD of zero displaced block and DBD_{adj} be the DBD of the block that has maximum block similarity to the current block. If $DBD_{(0,0)}$ is smaller than DBD_{adj}, the zero displacement is selected as the new search origin, otherwise, the location pointed to by the MV of an adjacent block with the maximum block similarity is used as the new search origin for BMA.

In the presented method, the possibility of falling into a local minimum can be reduced by using the motion direction of the object. If the adjacent blocks contain the same object as the current block, the block which contains the largest part of the same object is selected and the new search origin is set by its MV. By taking advantage of the correlations of adjacent blocks, the search area can be reduced, i.e., cutting down the number of matching candidate blocks with opposite motion direction.

2.2 *Search Window Adjustment*

After setting the new search origin, the size of search window of each block is determined by considering the block similarity and MV of the adjacent block.

BNSDOCID: <XP____771764A__I_>

The size of the search window for each block is determined at step 5 in Fig. 2. The maximum displacement of the MV is determined in some different ways according to the new search origin at step 4. When the new search origin is set as the position pointed to by the MV of the adjacent block ($DBD_{adj} < DBD_{(0,0)}$), the displacement is adjusted only considering the block similarity. In the DASW, we identify how the new search origin is well predicted using block similarity. Three classes of block, poor-, medium-, and well-predicted, are defined as follows:

The block is

$$= \begin{cases} \text{poor-predicted}, & \text{if block similarity} \leq T_{\text{low}}, \\ \text{medium-predicted}, & \text{if } T_{\text{low}} < \text{block similarity} \leq T_{\text{high}}, \\ \text{well-predicted}, & \text{if block similarity} > T_{\text{high}}, \end{cases} \tag{1}$$

where T_{low}, T_{high} are thresholds for classifying the block. The maximum displacements are set as w for the poor-predicted, $w/2$ for the medium-predicted, and $w/4$ for the well-predicted block, where w is an initial displacement of MV. For the well-predicted block, the current block is very similar to the adjacent block, and their motion displacement are also very similar. The best-matched block, therefore, is placed near the new search origin. This can be used to reduce the size of the search window. In the poor-predicted case, we can derive an opposite result.

Second, if the new search origin is set by zero displaced block ($DBD_{(0,0)} \leq DBD_{adj}$), the displacement is determined as follows:

displacement

$$= (\text{block similarity}) \times \text{Max}\{x,y \mid \text{MV component of the adjacent block}\} + w \times (1 - \text{block similarity}). \tag{2}$$

In this case, the motion vector lies between 0 and w, where (x,y) is MV for the adjacent block that has the largest block similarity. $DBD_{(0,0)} \leq DBD_{adj}$ means that the best matched candidate block may exist in the center-biased location.

For each block, the size of the search window is determined, and then the BMA is performed as a conventional one. This strategy can also be applied to some fast search algorithms such as TSS, NTSS, 2D-LOG, etc.

2.3 *Update the Thresholds*

In the DASW algorithm, three thresholds, T_{low}, T_{high} for classifying the block at the search window adjustment, and T_{seg} for image segmentation are used. Each threshold is closely related to the computational complexity and the performance of the proposed algorithm. The threshold T_{low} is used to classify the new search origin of the block into poor-predicted or medium-predicted and the threshold T_{high} is used for classifying the new search origin into medium-predicted or well-predicted. In our algorithm, we fixed the T_{low} and T_{high} to 30% and 70%, respectively. The thresholds T_{low} and T_{high} are derived from the relation between the block similarity and the differences of MV components of the two blocks. With some test sequences, we can find out that the blocks with high block similarity have very similar MVs of the blocks and estimated the thresholds T_{high} and T_{low} by considering block similarities and correlations of MVs. The threshold T_{seg} for image segmentation, is dynamically changed by considering the computational complexity, that is, the number of matching candidate blocks. If the computational complexity is higher than that of the expected number of matching blocks, the threshold is increased in proportion to the rate of increasing computational complexity such as

$$T_{\text{seg}}^{n} = T_{\text{seg}}^{n-1} + \bar{M} \times \frac{(\text{Number of matching blocks}) - (\text{Number of the expected matching blocks})}{(\text{Number of the expected matching blocks})} \tag{3}$$

where, $\bar{M}$ is the mean of the given frame. The initial threshold T_{seg} is set to a half of $\bar{M}$. With this updating policy, we can get MV with the expected computational complexity approximately.

3 Experimental Results

In the experiments, three video sequences (176 pels×144 lines and 30 frames/s) are tested. The two sequences, *Miss America* and *Susie*, contain a speaker with slow movements, which are typical in video phone or video conferencing. The sequence *Carphone* has a moderate motion field in automobile. The mean absolute error (MAE) distortion function is used as a matching criterion for BMA. A motion vector search is based on the luminance component with the search block of 16 by 16 with a displacement of 16. The MSE per pixel of prediction errors is taken as the measure of performance. The number of search points which is computed by counting the matching blocks for each block is used to compare the computational complexity of each method. Blocks in image boundary have a restrictive size of the search window in one or two directions. For example, a block located at (0,0) has the search window ranging from (0,0) to $(N+w, N+w)$ not from $(-w, -w)$ to $(N+w, N+w)$. With the three test sequences, we have compared the performance of the search window adjustment methods—a conventional method (conv.) where

BNSDOCID: <XP____771764A__I_>

Table 1 Average MSE per pixel and the number of search points (NSP) per block for each method applied to FSBMA with maximum motion displacement $w=16$.

	Miss America		Susie		Carphone	
BMA method	MSE	NSP	MSE	NSP	MSE	NSP
FSBMA-conv.	6.96	886	44.62	886	66.10	886
FSBMA-AASW	7.29	415	48.15	483	70.06	496
FSBMA-DASW	7.02	376	44.76	462	67.24	498

Table 2 Average MSE per pixel and the number of search points (NSP) per block for each method applied to TSS with maximum motion displacement $w=16$.

	Miss America		Susie		Carphone	
BMA method	MSE	NSP	MSE	NSP	MSE	NSP
TSS-conv.	16.69	28	65.29	29	92.54	29
TSS-AASW	13.17	32	58.52	32	87.61	32
TSS-DASW	11.22	18	56.25	19	85.19	20

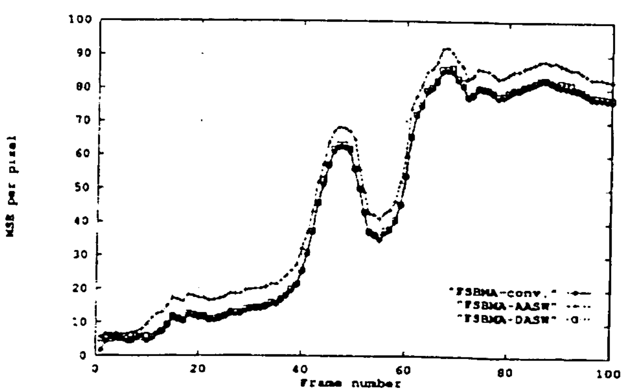
Table 3 Average MSE per pixel and the number of search points (NSP) per block for each method applied to NTSS with maximum motion displacement $w=16$.

	Miss America		Susie		Carphone	
BMA method	MSE	NSP	MSE	NSP	MSE	NSP
NTSS-conv.	11.42	17	57.76	19	76.76	18
NTSS-AASW	11.56	21	58.22	21	76.73	20
NTSS-DASW	9.42	17	53.87	18	71.92	18

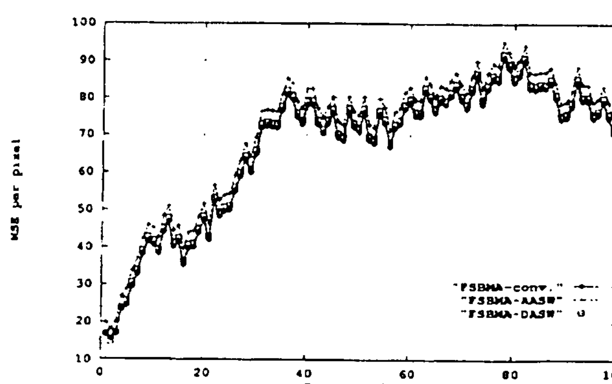
the size of the search window is set by initial displacement, the scheme AASW proposed by Feng *et al.*,[13] and proposed scheme DASW—which are applied to FS, TSS, and NTSS.

The average performances are presented in Tables 1, 2, and 3 with the three test sequences. In FSBMA with the threesearch window adjustment methods, the proposed scheme DASW has fewer search points by about 50% than conventional FS without great loss of MSE performance. Also, compared with the AASW scheme proposed by Feng *et al.*,[13] the DASW approach has better MSE performance with a similar number of search points. Table 2 shows the average performances for TSS. From the table, the efficiency of the DASW is more clear than the other two schemes in the number of search points and the MSE performance. The number of search points is reduced by 30% to 40%, and the MSE per pixel is better by about 3% to 10%. With the *Susie* sequence, the performance gain is noticeable. In NTSS, the performance gain is smaller than that of TSS because NTSS has been developed based on the center-biased motion vector in smoothly varying image sequences. The number of search points of the DASW method is similar to that of the original NTSS, but the DASW shows a better MSE performances with the three video sequences. The performances of the MSE and the number of search points are better than those of AASW. From the three video sequence, the performance gain is great in a video sequence with some moderate motion fields rather than that with the stationary motion field such as the *Miss America* sequence.

Detailed MSE performances per pixel of the two sequences *Susie* and *Carphone* are shown in Figs. 3, 4, and 5. In our scheme, an object with high motion is very well predicted since block similarity is calculated from the object-based region which is preprocessed for BMA. Also, falling into a local-minimum is overcome by using the object-based motion structures.



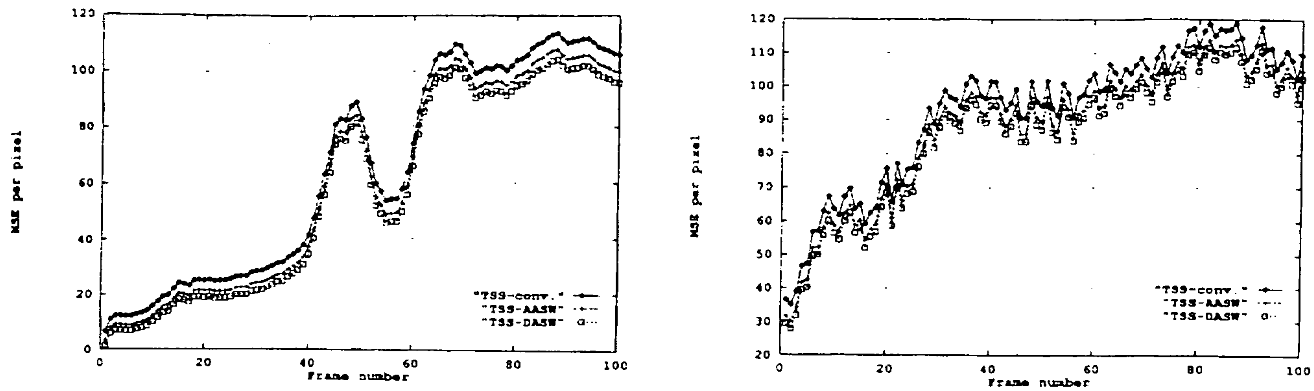
(a) Susie



(b) Carphone

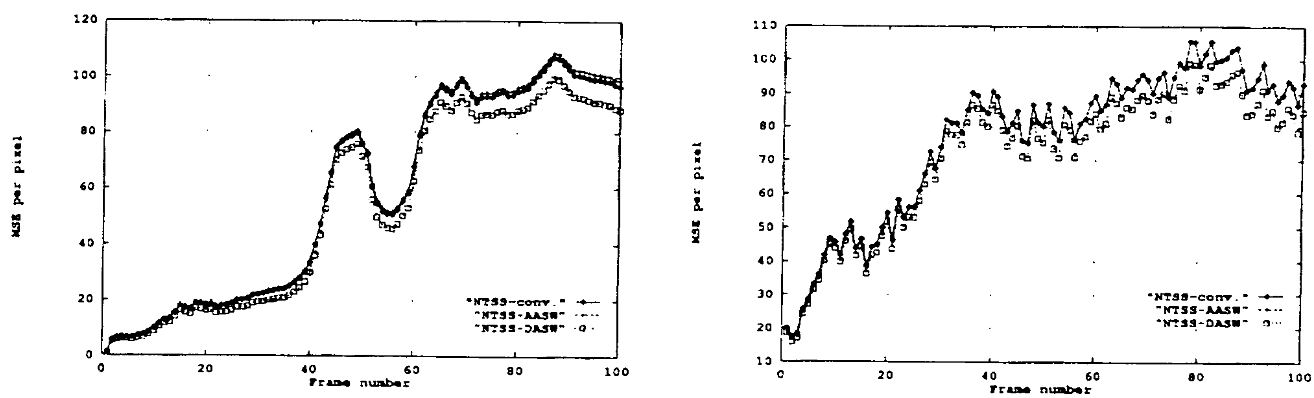
Fig. 3 MSE performances of the three methods applied to FS with maximum displacement $w=16$.

BNSDOCID: <XP____771764A__I_>



(a) Susie (b) Carphone

Fig. 4 MSE performances of the three methods applied to TSS with maximum displacement $w=16$.



(a) Susie (b) Carphone

Fig. 5 MSE performances of the three methods applied to NTSS with maximum displacement $w=16$.

As the simulation results have shown the low computational complexity of the proposed method, it can be used for low complexity video coding applications such as video phone or video conferencing.

4 Conclusion

In this paper, we propose a dynamic search window adjustment technique for BMA to reduce the computational complexity of FSBMA, and to overcome the problem of falling into the local minimum in several fast search algorithms. The proposed method uses a block similarity and a DBD to adaptively adjust the size of the search window. The technique also can be easily applied to FSBMA and several fast search algorithms to get more efficiency. The experimental results have shown that the MSE performance and the reduction of the number of search points with the proposed DASW scheme are better than those of conventional and the AASW. The simulation results have proved the DASW scheme can be used for low complexity video codec applications such as video telephony or video conferencing.

References

1. D. L. Gall, "MPEG: A video compression standard for multimedia application," *Commun. ACM* **34**(4), 46–58 (1991).
2. M. Liou, "Overview of the Px64 kbit/s video coding standard," *Commun. ACM* **34**(4), 59–63 (April 1991).
3. R. Schaphorst, "Draft recommendation H.263 (Video coding for low bitrate communication)," *ITU-T SG15 WP15/1* (July 1995).
4. Y. Baek, H.-S. Oh, and H.-K Lee, "Block-matching criterion for efficient VLSI implementation," *Electron. Lett.* **32**(13), 1184–1185 (1996).
5. J. R. Jain and A. K. Jain, "Displacement measurement and its application in interframe image coding," *IEEE Trans. Commun.* **Com-29**(12), 1799–1808 (1981).
6. R. Li, B. Zeng, and M. L. Liou, "A new three-step search algorithm for block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.* **4**(4), 438–441 (1994).
7. T. Koga, K. Linuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion-compensated interframe coding for video conferencing," *Proc. NTCSI*, G5.3.1–G5.3.5 (Dec. 1981).
8. R. Srinivasan and K. R. Rao, "Predictive coding based on efficient motion estimation," *IEEE Trans. Commun.* **33**(8), 888–896 (1985).
9. M.-J. Chen, L.-G. Chen, and T.-D. Chiueh, "One-dimensional full search motion estimation algorithm for video coding," *IEEE Trans. Circuits Syst. Video Technol.* **4**(5), 504–509 (1994).
10. M. Ghanbari, "The cross-search algorithm for motion estimation," *IEEE Trans. Commun.* **38**(7), 950–953 (1990).
11. L.-G Chen, W.-T Chen, Y.-S Jehng, and T.-D. Chiueh, "An efficient parallel motion estimation algorithm for digital image processing," *IEEE Trans. Circuits Syst. Video Technol.* **1**(4), 378–385 (1991).
12. L.-W Lee, J.-F Wang, J.-Y. Lee, and J.-D. Shie, "Dynamic search-window adjustment and interlaced search for block-matching algorithm," *IEEE Trans. Circuits Syst. for Video Technol.* **3**(1), 85–87 (1993).
13. J. Feng, K.-T. Lo, H. Mehrpour, and A. E. Karbowiak, "Adaptive block matching motion estimation algorithm for video coding," *Electron. Lett.* **32**(18), 1542–1543 (1995).

BNSDOCID: <XP____771764A__I_>

14. S. Kappagantula and K. R. Rao, "Motion compensated interframe image prediction," *IEEE Trans. Commun.* Com-33(9), 1011–1015 (1985).
15. B. Liu and A. Zaccarin, "New fast algorithms for the estimation of block motion vectors," *IEEE Trans. Circuits Syst. Video Technol.* 3(2), 148–157 (1993).
16. V. Seferidsi and M. Ghanbari, "General approach to block-matching motion estimation," *Opt. Eng.* 32(7), 1464–1474 (1993).
17. I. Pitas, *Digital Image Processing Algorithm*, pp. 254–297, Prentice-Hall, Englewood Cliffs, NJ (1993).
18. E. Chan, A. A. Rodringuez, R. Gandhi, and S. Panchanathan, "Experiments on block-matching techniques for video coding," *ACM Multimedia Syst.* 2(5), 228–241 (1994).
19. W. Li and E. Salari, "Successive elimination algorithm for motion estimation," *IEEE Trans. Circuits Syst. Video Technol.* 4(1), 105–107 (1995).
20. Y.-L. Chan and W.-C. Siu, "New adaptive pixel decimation for block motion vector estimation," *IEEE Trans. Circuits Syst. Video Technol.* 6(1), 113–118 (1996).
21. L.-M. Po and W.-C. Ma, "A novel four-step search algorithm for fast block motion estimation," *IEEE Trans. Circuits Systems Video Technol.* 6(3), 313–317 (1996).
22. H. Jozawa, K. Kamikura, A. Sagta, H. Kotera, and H. Watanabe, "Two-stage motion compensation using adaptive global MC and local affine MC," *IEEE Trans. Circuits Syst. Video Technol.* 7(1), 75–85 (1997).

Hwang-Seok Oh received his BS degree in computer science from Kyungpook National University, Korea, in 1992 and his MS degree in computer science from Korea Advanced Institute of Science and Technology (KAIST), Korea, in 1994. Currently he is a PhD candidate in the Department of Computer Science at KAIST, Korea. He is a member of IEEE, ACM, and SPIE. His research interests include video coding, multimedia systems, and VLSI architectures for signal processing.

Heung-Kyu Lee received his BS degree in electronics engineering from the Seoul National University, Seoul, Korea, in 1978, and MS and PhD degrees in computer science from the Korea Advanced Institute of Science and Technology (KAIST), in 1981 and 1984, respectively. From 1984 to 1985 he served as a post-doc at the University of Michigan, Ann Arbor. In 1986 he joined the Korea Advanced Institute of Science and Technology as a professor. Since 1990, he has been a research staff member at the Satellite Research Center in KAIST. He is a member of IEEE and ACM. His major interests are VLSI and parallel processing, fault-tolerant and real-time computing, and multimedia systems.

BNSDOCID: <XP____771764A__I_>

THIS PAGE BLANK (USPTO)